# LAB: Modeler Flows in IBM Watson Studio (Updated)

For this lab, you will use the IBM Cloud Account and the Watson Studio project you created when you worked on the Lab in the previous lesson titled *Creating a Watson Studio Project with Jupyter Notebooks*. To create a free Watson Studio account, use the following link https://cloud.ibm.com/catalog/services/watson-studio
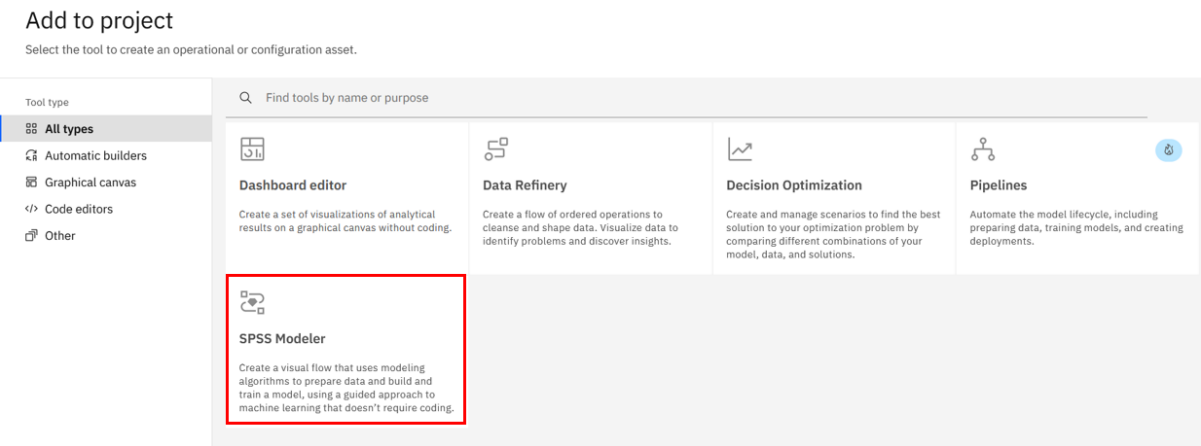
Objective for Exercise:

- Load an example flow, run it, and examine results.
- Add an Auto Numeric model, run it, and examine results.
- Get predictions for new cases using a model we built.
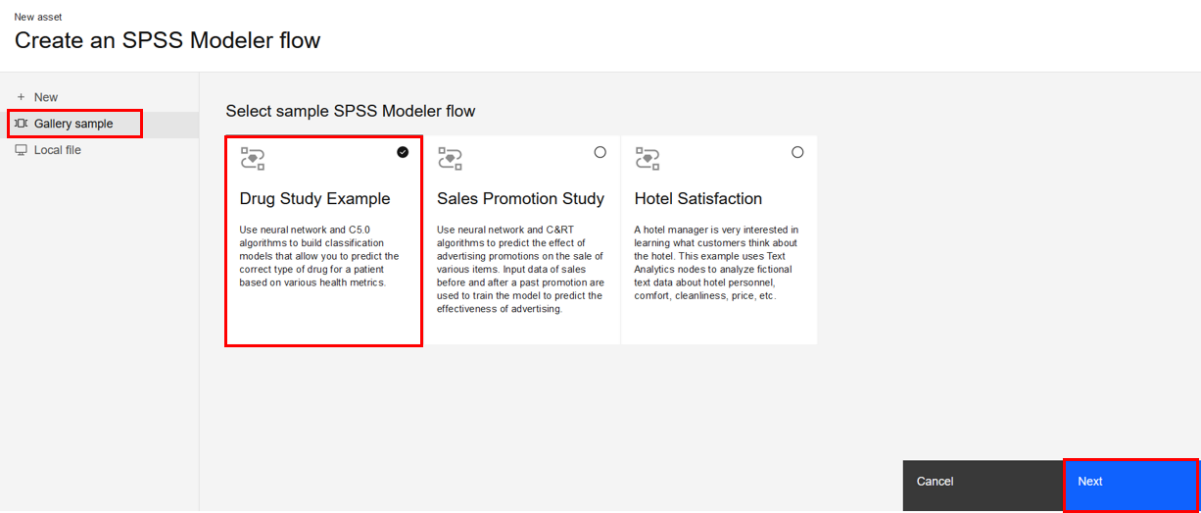
Step 1 - Open your project in Watson Studio, then click **Assets** tab and click **Add to Project**
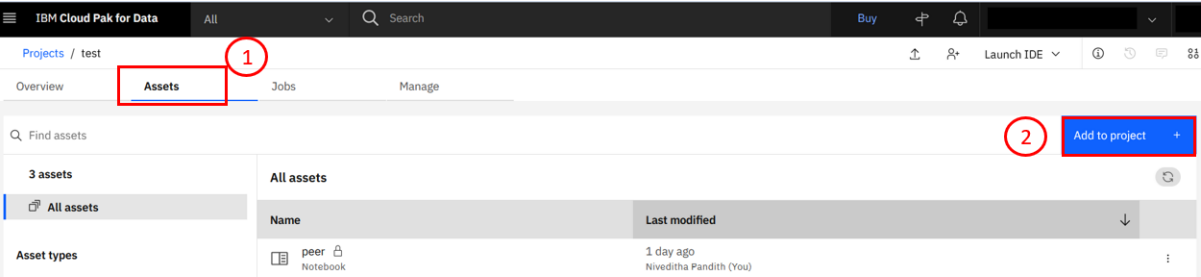
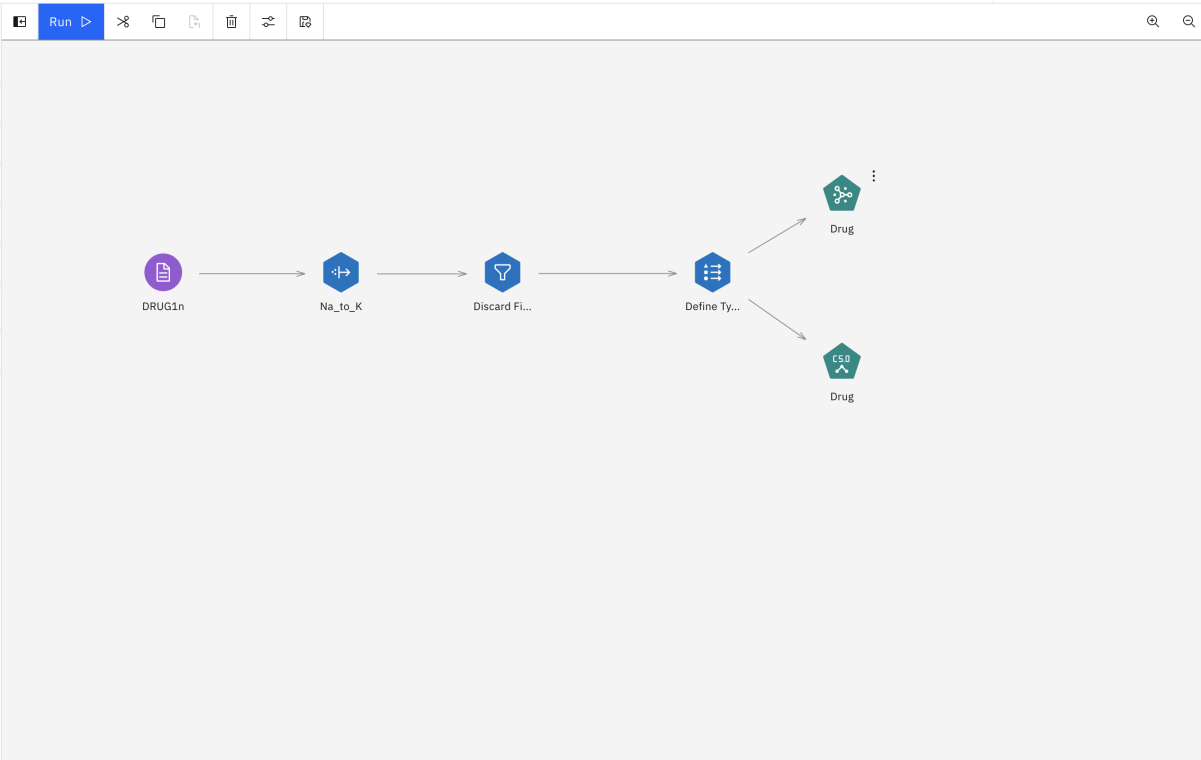Scroll down and select **SPSS Modeler**.

Step 2 - Next, select **Gallery Sample** and select **Drug Study Example**:

Step 3 - Selecting the Drug study example on the left would give us the flow we have already seen in the previous section, then click "Create" button in the lower right corner.

Once the flow loads, you will see an image similar to this:

Step 4 - Reading from left to right, we can examine the flow. The purple circle corresponds to the data set of some fictional sales, the next node is deriving a field Increase that will be our target variable. It is based on the increase of sales. Then we see a type node which specifies our target variable and predictors. In this case our target variable is numeric and continuous. Finally, there are two models - a neural network and a C&RT decision tree. We can run the flow as is, or we can modify it to see other possibilities.

Step 5 - First, let's see what we can get with the existing models. Click **Run** button on the top of the canvas:

Step 6 - After some execution time, you'll get dark gold model nuggets for the two models. Clicking on three dots in the right side of one of them you can select "View model" and examine the information.
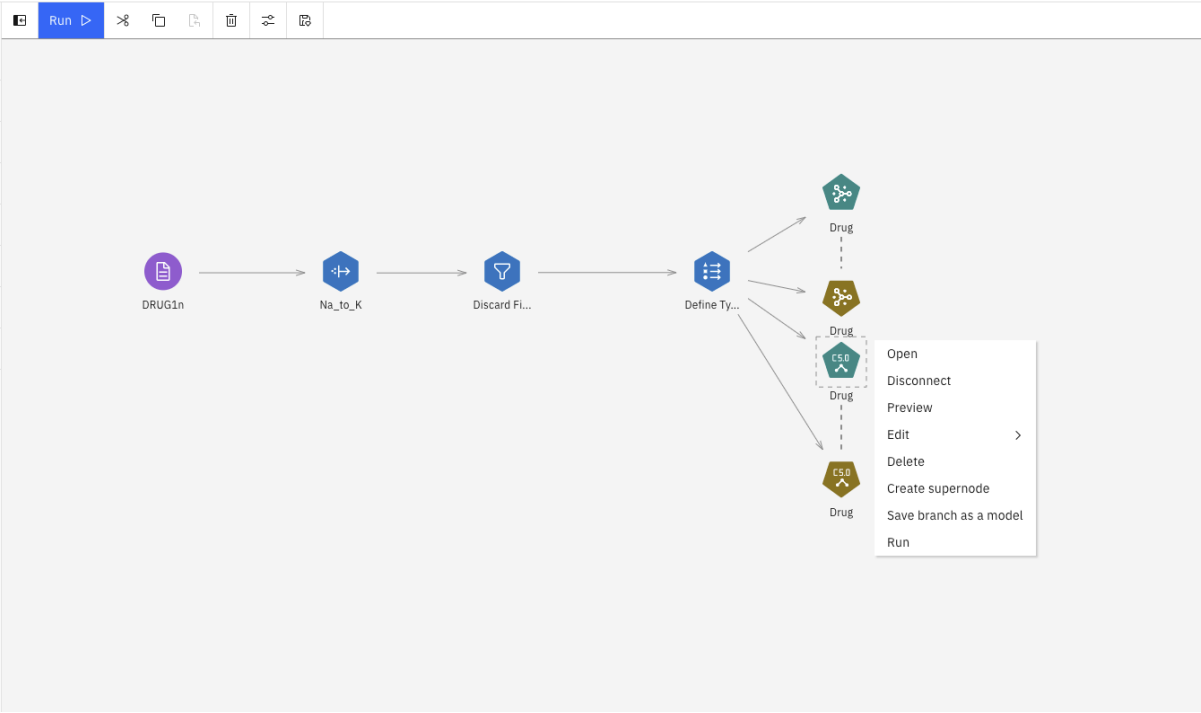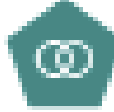
For example, the neural network diagram looks like this:
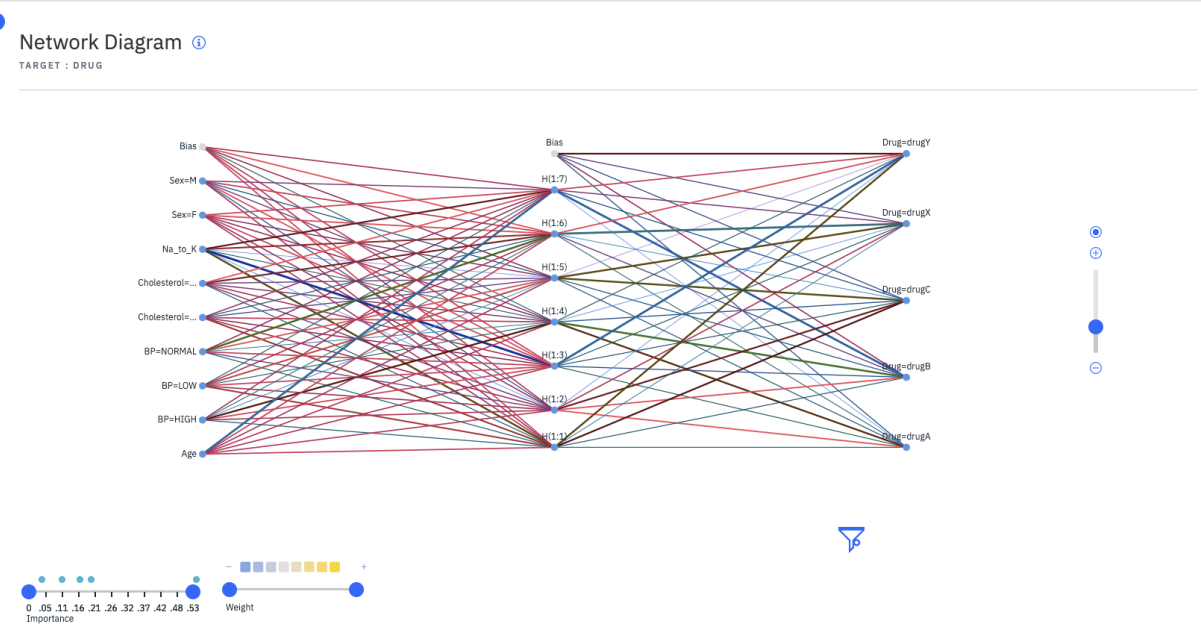
Step 7 - Now, let's add the "Auto-numeric" model to the flow. Open the modeling palette on the left side by clicking on Modeling:

- Auto Classifier
- Auto Numeric
- Auto Cluster
- Bayes Net
- C5.0
- C&R Tree
- CHAID
- Quest
- Tree-AS
- Random Trees

Random Forest

Decision List

Step 8 - In the resulting modeling palette, pick the Auto-Numeric model:
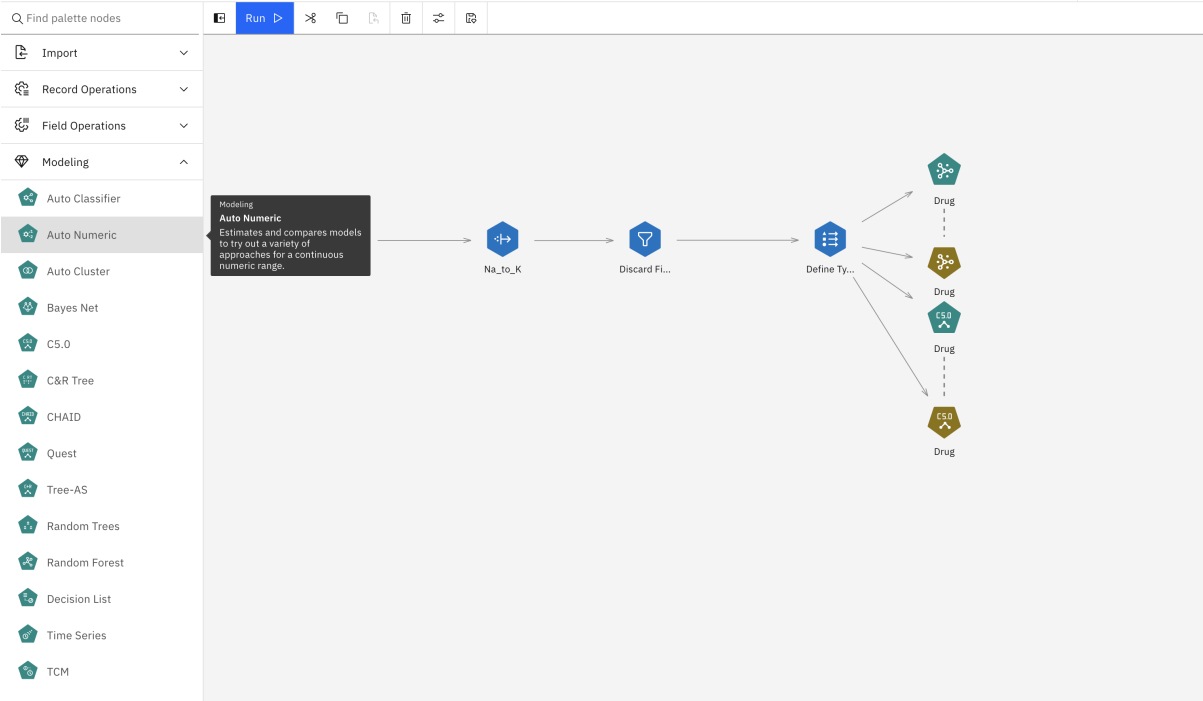
Step 9 - Click on the Auto Numeric node and while keeping the left mouse button pressed, drag it onto the canvas.
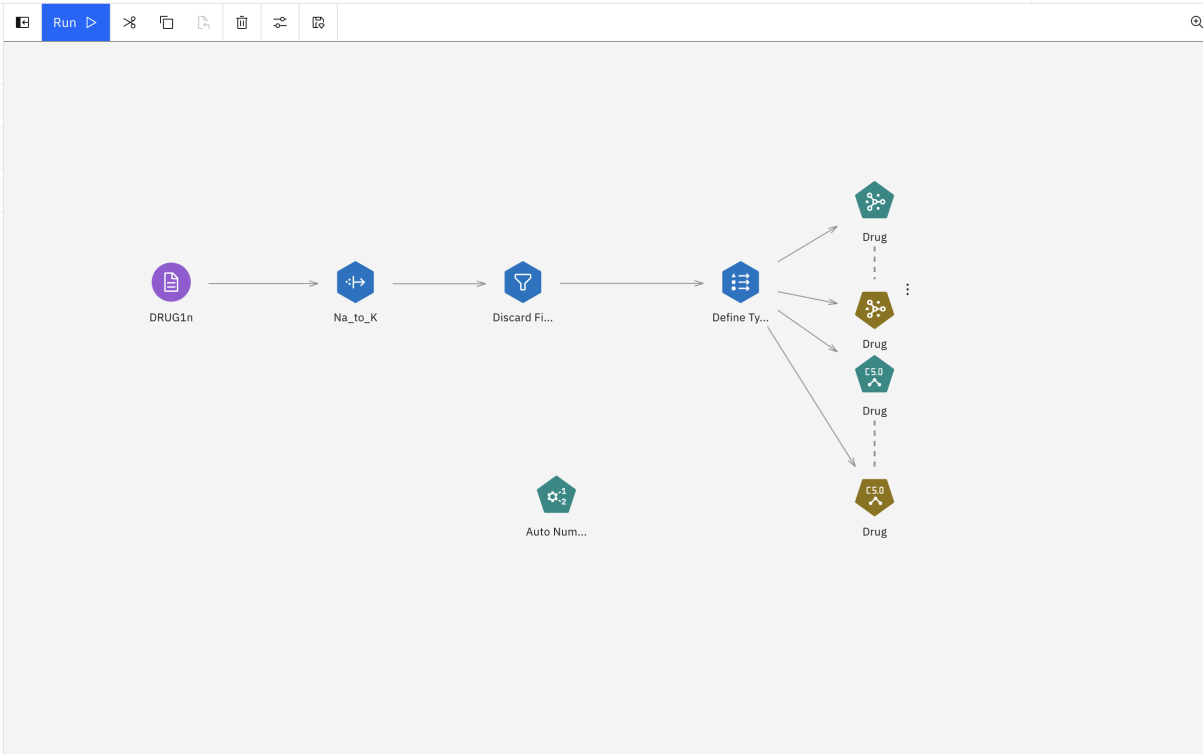
Step 10 - Now, hover over the type node with your mouse and click on the circle with an arrow that appears on the right side. Make a connection to the dot on the left side of the Auto Numeric node.
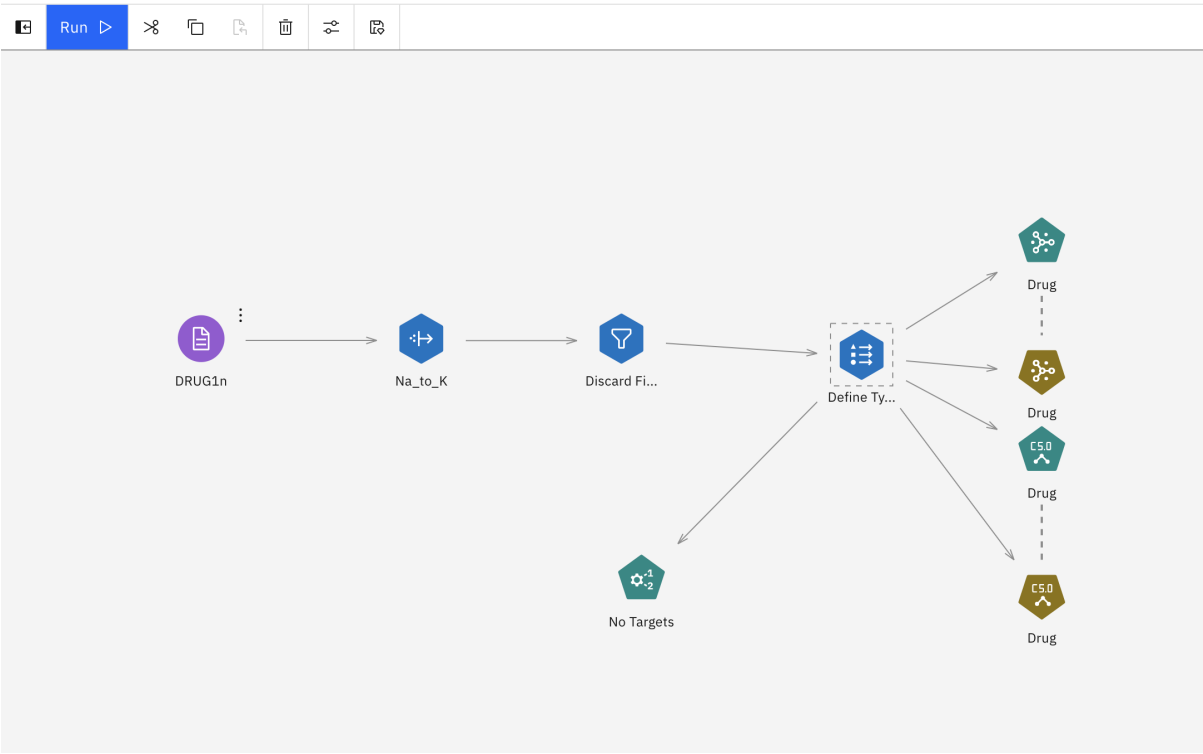
Step 11 - Double-click the "Auto-numeric" and click "Use custom field roles", use the drop-down to chose a target, we will use "Na_to_K". Click "Add Columns" next to Inputs to add inputs.

Fields

☑ Use custom field roles ⓘ

Target

[... ▼]

Inputs

Remove 🗑   Add Columns +

☑	**Field Name**

Partition

[... ▼]

Splits

Remove 🗑   Add Columns +

☑	**Field Name**

Cancel	Save

Step 12 - Check all. the fieldswe want to use for the input and click "OK" then "Save"

Select Fields for Na_to_K

	Field Name	Data Type
☑	Age	# integer
☑	Sex	abc string
☑	BP	abc string
☑	Cholesterol	abc string
☑	Drug	abc string

Filter: # abc   Reset ↺

Find in column Field name

Cancel   OK

Step 13 - Finally, run the new branch of the flow by clicking on the three dots on the upper right part of Auto Numeric node and selecting the last option in the menu, "Run".
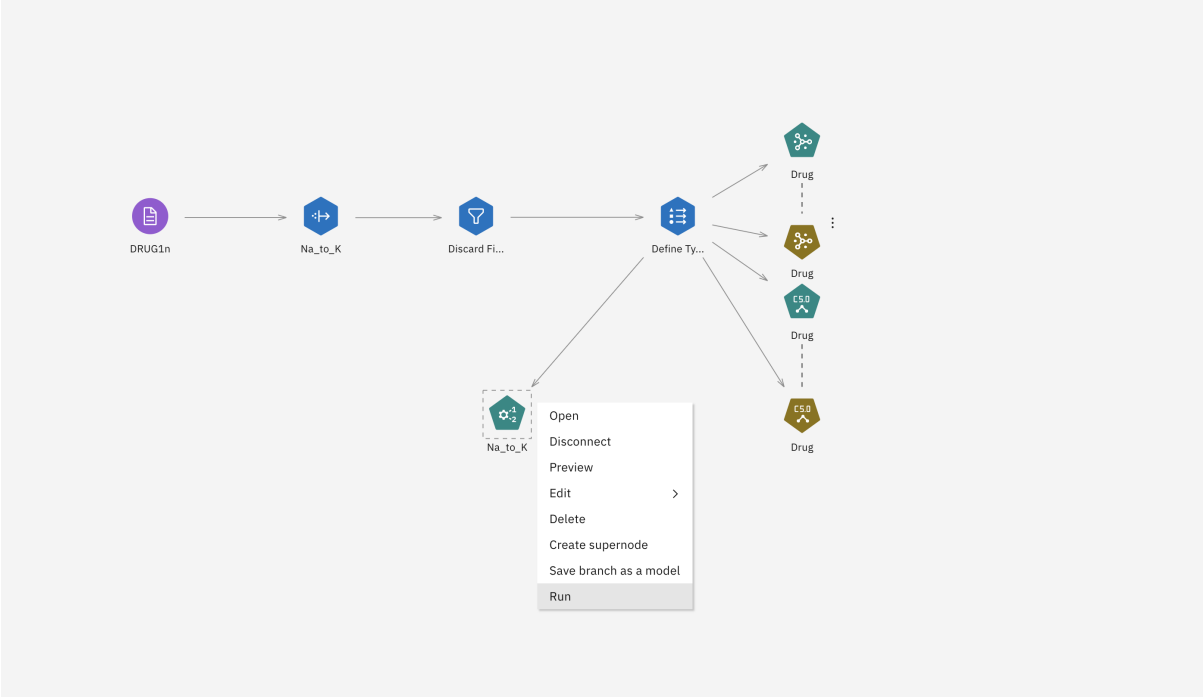
Step 14 - After the execution ends, we get a model nugget:
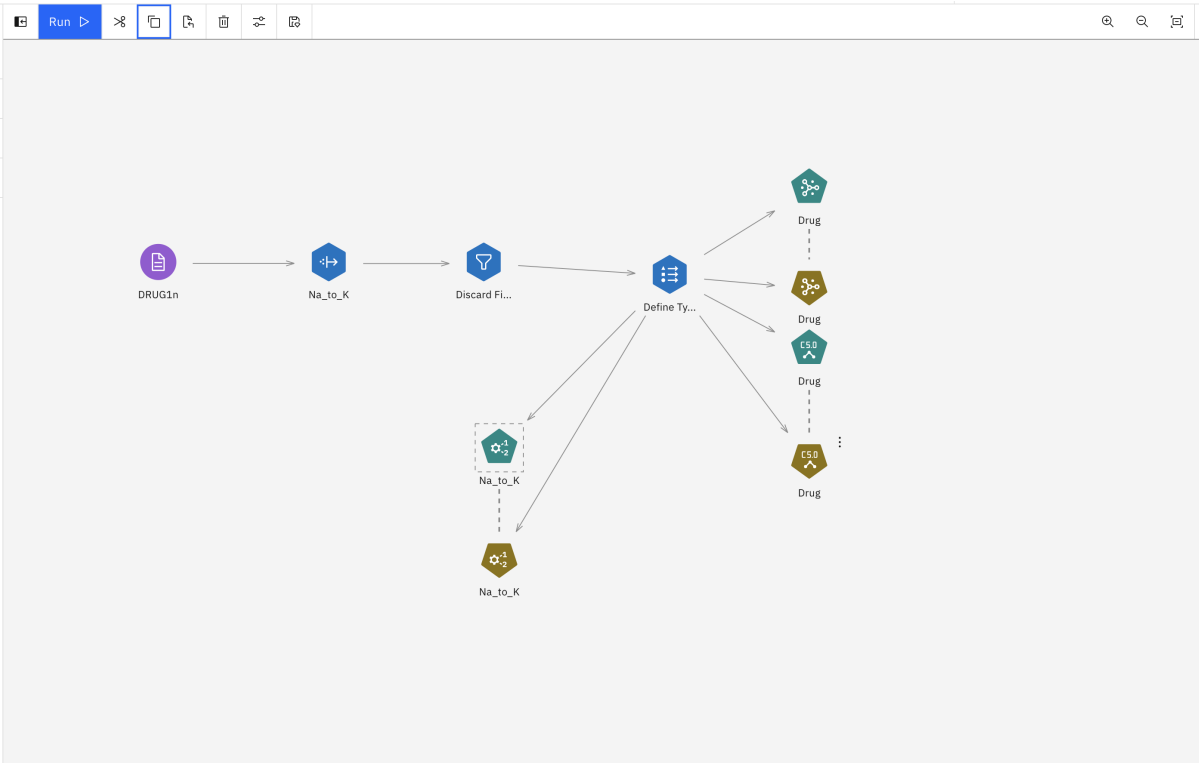
and viewing the model, we see the following table:

Auto Numeric - Models ⓘ

TARGET : NA_TO_K

USE	ESTIMATOR	ACCURACY ▼	RELATIVE ERROR	BUILD TIME (MINS)	NO. FIELDS USED	ACTIONS
☑	XGBoost Tree 1	0.941	0.134	< 1	5	🗑
☑	Random Trees	0.859	0.290	< 1	5	🗑
☑	C&RT	0.838	0.300	< 1	5	🗑
☑	Generalized Linear 1	0.809	0.346	< 1	5	🗑
☑	LE	0.808	0.347	< 1	5	🗑

This tells us that five different models have been built, and also some properties of the models.

XGBoost is a very popular model, representing gradient-boosted ensemble of decision trees. The algorithm was discovered relatively recently and has been used in many solutions and winning data science competitions. In this case, it created a model with the highest accuracy, which "won" as well. "C&RT" stands for Classification and Regression Tree", a decision tree algorithm that is widely used. This is the same decision tree we saw earlier when we built it separately. "LE" is "linear engine", an IBM implementation of linear regression model that includes automatic interaction detection. The model coefficients are shown in **Parameter Estimates** table. We can see that several coefficients correspond to a combination of one category of variable *class* with continuous variable *Promotion*. This is called "an interaction effect". You will not see such features in simple linear regression models.

← Auto Numeric - Na_to_K

Linear Regression ⓘ ✕

EVALUATION

Model Evaluation

Observed by Predicted

Residuals by Predicted

Normal P-P Plot of Residuals

Residual Histogram

MODEL VIEWER

Model Information

Records Summary

Feature Importance

Tests of Model Effects

Parameter Estimates

Model Evaluation ⓘ

TARGET : NA_TO_K

Model Accuracy

0.661

Model Evaluation Measures

R^2	0.661
Adjusted R^2	0.646

Step 15 - Now you can experiment with Modeler flows on your own. You can use various nodes, just don't forget to put a Type node before any modeling node and to specify your inputs and targets.

There are several other models to predict a continuous target, for categorical targets, you can try Logistic Ragression, Support Vector Machines etc.

You will learn more about various models when you take a course on Machine Learning.

Author(s)

Aije Egwaikhide

Changelog

Date	Version	Changed by	Change Description
2021-01-07	2.1	Aije	Created new example and updated screenshuots
2020-08-25	2.0	Lavanya	Migrated Lab to Markdown and added to course repo in GitLab